

Amendments to the Specification:

Please replace the title beginning on page 1, line 1, with the following amended title:

~~SYSTEMS, METHODS, AND COMPUTER PROGRAM PRODUCTS~~ BYPASSING
DISK I/O OPERATIONS TO IMPROVE THE PERFORMANCE OF PORTED
APPLICATIONS, SUCH AS A DATABASE

Please replace the paragraph beginning on page 1, line 7, with the following amended paragraph:

This patent application is a divisional application of and claims priority to ~~co-pending~~
~~and~~ commonly-assigned U.S. Patent No. 6,754,734 ~~Application No. 10/033,810~~, titled “Systems,
Methods, and Computer Program Products to Improve Performance of Ported Applications, Such
as a Database” filed on December 18, 2001, by David H. Goode and William E. Malloy, which is
incorporated herein by reference in its entirety.

Please replace the paragraph beginning on page 1, line 24, with the following amended paragraph:

Typically complex computer applications, such as a database, are ported to a 25 variety
of computer systems. The porting process often includes special changes to the application to
enable efficient and complete operation of the application on different computer systems. I/O
operations are a significant factor in the overall performance of a complex computer application.
High-performance computer applications, such as a database, may issue asynchronous, direct
disk I/O commands which are not supported on the target system. A “target computer system” as
used herein refers to a computer system environment consisting of one or more specific
programming languages, the application programming interfaces (APIs) available in the
programming languages, and the associated file system or file systems. Therefore, changes to I/O
operations may be made during the porting of an application to ensure efficient operation of the
application on the computer system. Such a computer system may include the products sold
under the trademarks IBM® S/390® (hereinafter S/390) that includes the IBM® OS/390®
(hereinafter OS/390) operating system and associated disk volumes.

Please replace the paragraph beginning on page 2, line 9, with the following amended paragraph:

Disk volumes are units of data storage that typically include data and the information used to access and manipulate the data. Disk volumes may be used to store a file system and information necessary to manipulate the file system. For example, when implementing database applications that may include disk I/O access commands for operation on the IBM OS/390 that supports UNIX® System Services (hereinafter OS/390 UNIX) the facilities of a hierarchical file system (HFS) may be employed. However, file systems, such as the OS/390 UNIX HFS, may only support queued disk I/O access and minimal I/O caching.

Please replace the paragraph beginning on page 13, line 21, with the following amended paragraph:

The system-dependent code module 111 isolates system-specific code to a system-dependent layer. Therefore, when porting a database management tool or other application 108 to another operating system 110, such as porting a database from an AIX® operating system to OS/390 UNIX, the changes may be isolated to the system-dependent code module 111. It will be appreciated that the present invention may operate without support of the system-dependent code module 111 or the low-level direct I/O interface 113, or the I/O subsystem 114 and may interface directly with the disk volume 104.

Please replace the paragraph beginning on page 21, line 6, with the following amended paragraph:

As shown in Figure 4B, and in element ~~418~~420, the method of the present invention advantageously exploits the performance characteristics of disks 122 that perform I/O operations faster when given fewer I/O requests 121. That is as shown in element 420, ordered computer code having at least one asynchronous direct I/O access command and being located in a loop is identified. Next, in element 422, loops are processed in the application program code 108, such as database code, in which a plurality of identified I/O requests 230 are executed. ~~It will be appreciated that the functions described with respect to element 404 in Figure 4A may be used to locate the identified I/O requests 230.~~ The loops are likely to appear in most database

applications in the areas of the computer code that are responsible for moving data 160 from the cache memory 558 (as shown in Figure 5) to the disk 122.